

Multiple fundamental frequency estimation based on sparse representations in a structured dictionary ^{☆,☆☆}

Michal Genussov, Israel Cohen ^{*}

Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:

Available online 11 September 2012

Keywords:

Piano transcription
Music information retrieval
Sparse representations
Multi-pitch estimation

ABSTRACT

Automatic transcription of polyphonic music is an important task in audio signal processing, which involves identifying the fundamental frequencies (pitches) of several notes played at a time. Its difficulty stems from the fact that harmonics of different notes tend to overlap, especially in western music. This causes a problem in assigning the harmonics to their true fundamental frequencies, and in deducing spectra of several notes from their sum. We present here a multi-pitch estimation algorithm based on sparse representations in a structured dictionary, suitable for the spectra of music signals. In the vectors of this dictionary, most of the elements are forced to be zero except the elements that represent the fundamental frequencies and their harmonics. Thanks to the structured dictionary, the algorithm does not require a diverse or a large dataset for training and is computationally more efficient than alternative methods. The performance of the proposed structured dictionary transcription system is empirically examined, and its advantage is demonstrated compared to alternative dictionary learning methods.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Transcription of music is defined as the process of identifying the parameters of an acoustic musical signal, which are required in order to write down the score sheet of the notes [1]. One of the most important parameters is the *pitch*, which is represented in written music by the note symbol. For convenience, we shall refer here to the task of pitch identification as “transcription”. Automatic transcription of music is important, since it allows structured audio coding, it is a helpful tool for musicians, enables modifying, rearranging and processing music in an efficient way, and is the basis for interactive music systems.

We need to separate between two different cases – transcription of monophonic music and transcription of polyphonic music. Monophonic music is the case in which a single note is played at each time instant. For this case, automatic transcription is practically a solved problem. Several proposed algorithms are reliable, commercially applicable and operate in real time. However, transcription of polyphonic music, in which more than one note is played at a time, is much more complicated. To the best of our

knowledge, no existing algorithm can identify multiple pitches in an accuracy close to 100%. This is somehow not intuitive to understand, since when a trained human (such as a musician) listens to a polyphonic music piece, he can distinguish between the different notes and identify them, although played simultaneously. The difficulty stems from the fact that most often, especially in western music, several harmonics of different notes overlap. This causes a problem in assigning the harmonics to their true fundamental frequencies, and in deducing spectra of several notes from their sum [2,3].

Since the 1970s, when Moorer built a system for transcribing duets [4], there has been a growing interest in transcribing polyphonic music. The offered algorithms can be divided into three main groups: time-based, frequency-based and time-frequency-based algorithms. The time-based group includes methods which are based on the autocorrelation function [5–7] and on the periodicity of the signal [8–10]. The frequency-based group includes methods which are based on typical harmonic patterns in the frequency domain [11,12], and which can be mapped to a logarithmic scale to better fit the human auditory system [13–16]. The combined time-frequency-based group includes methods which use a time-frequency image, such as a spectrogram or a scalogram [17], or cepstrum analysis [18,19].

Some of the above methods [11,19] are based on the assumption that the spectral shape of the harmonics can be modeled by a constant function, therefore spectrums of several notes can be deduced from their combination. This is very inaccurate, since the spectral shape changes as a function of many factors, which include the type of the musical instrument, the total intensity of the

[☆] This research is part of the M.Sc. thesis of the first author, M. Genussov, “Transcription and classification of audio data by sparse representations and geometric methods”, Technion, Israel Institute of Technology, 2010.

^{☆☆} This research was supported by the Israel Science Foundation (grant no. 1130/11).

^{*} Corresponding author.

E-mail addresses: michalgenus@gmail.com (M. Genussov), icohen@ee.technion.ac.il (I. Cohen).

tone, the fundamental frequency and the stage in the time envelope of the sound, i.e. the ADSR (attack, decay, sustain, release) envelope of the note [20]. Other methods [10,12,15,21] are limited by the fact that they are supervised methods, i.e., require a training set of pieces in order to transcribe another musical piece.

The idea of using *Sparse representations* as a time-based or frequency-based method for transcription of polyphonic music, was first suggested by Abdallah and Plumbley [22]. It was later improved and expanded [8,23,24], and inspired other works [25–27].

The term “Sparse representations” means writing a signal as a linear combination of very few underlying functions, which are contained in a *dictionary* of underlying functions. This is implemented by multiplying the dictionary of the underlying functions by a *sparse* vector (i.e., a vector that contains very few non-negative elements compared to its length), giving the method its name.

The motivation for applying sparse representations of music signals, is that in played music, only a small number of notes is played simultaneously compared to the number of available notes. In the frequency domain, the approach is based on the idea that power spectra of different notes approximately add, assuming random phase relationships. However, the existing algorithm for transcription of music using sparse representations, developed by Abdallah and Plumbley [24], suffers from a major drawback – it requires a large and diverse database of notes, in order to build a representative and reliable dictionary for the transcription. For example, if a note is played only in combination with other notes, the learned dictionary will not represent the individual note, but the combination in which it resides.

Several researchers, such as Raczynski et al. [28], Hennequin et al. [29], Ewert and Müller [30] and Vincent et al. [31–33], used Non-negative Matrix Factorization (NMF) for transcription of polyphonic music. Some of their work involved representing the signal using an adaptive structured decomposition, which is quite similar to our approach. However, while they used NMF for the decomposition, we used sparse representation and structured dictionary learning, which involve different updating of the decomposed parameters. A more detailed comparison is introduced in the Conclusions.

In this paper, we present an algorithm for multiple pitch estimation based on sparse representations in a structured dictionary, suitable for the spectra of music signals. It achieves good estimation results even with a small and limited database of notes, thus overcoming the drawback of the former algorithm based on sparse representations. We show the advantages of our algorithm over other algorithms for transcription of polyphonic music, using experiments on synthesized and recorded piano music.

The structure of the paper is as follows: In Section 2, we describe the musically-structured algorithm, based on sparse representations and spectral analysis of music signals. In Section 3, we review the implementation of the algorithm in a transcription setup for polyphonic music. In Section 4, experimental results are presented, and the algorithm is compared to other transcription methods. In Section 5, we conclude the paper and suggest future expansions.

2. Musically-Structured (MS) sparse representations

Our transcription algorithm is based on sparse representations with our new parametric dictionary, suitable for the spectra of music signals. We define this dictionary as the *Musically-Structured dictionary*. The algorithm is composed of two stages: Sparse coding and structured dictionary learning.

2.1. Sparse coding

The purpose of sparse coding is to approximate the solution of the following (P_0) problem:

$$(P_0): \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (1)$$

The vector \mathbf{x} is a sparse vector, encoding the columns (atoms) of the dictionary \mathbf{A} , such that together they represent the signal \mathbf{y} . This method was shown useful to capture the meaningful characteristics of signals, such as images and audio [34].

An error-tolerant modified version for multiple signals:

$$(P_0^\varepsilon): \min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 < K, \quad 1 \leq i \leq M, \quad (2)$$

where F is the Frobenius norm, and \mathbf{x}_i are the columns of \mathbf{X} . In our case, each of the columns of the matrix \mathbf{Y} , i.e., the signals, is the Constant Q Transform (CQT) [35], at a certain time window. We use the CQT instead of the Short-Time Fourier Transform (STFT), since it is more suitable for the auditory system. The STFT can be modeled as a bank of linearly spaced filters, whose bandwidths are constant. On the other hand, the CQT can be modeled as a bank of *logarithmically* spaced filters, whose bandwidths are logarithmically increasing as a function of frequency. The Q parameter, which is the bandwidth to center-frequency ratio, is constant as a function of the frequency. In the original CQT transform, the length of the time window decreases as a function of frequency. We use a degenerated version of it, with a constant window length. Because of its logarithmic frequency scale, the CQT better fits the auditory system than the STFT [35], and also reduces the computational complexity by using less frequency bins than the STFT.

The matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ is the Musically-Structured dictionary in which each atom (column) is the CQT of a single note. Each column in $\mathbf{X} \in \mathbb{R}^{m \times M}$ encodes a linear combination of the notes from \mathbf{A} which are played in the time window of the corresponding column in $\mathbf{Y} \in \mathbb{R}^{n \times M}$. At each time window, the corresponding column in \mathbf{X} is determined as the sparsest column which minimizes the Frobenius norm $\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F$. In our algorithm we use a greedy method for sparse coding, since it allows to pre-define the cardinality (the number of the non-zero elements K) of the sparse vector, according to the evaluated number of notes at each time window. Specifically, we choose to use the OMP algorithm [36,37], which settles a good compromise between complexity and performance, compared to other algorithms for sparse coding, such as Matching Pursuit (MP) or Basis Pursuit (BP) algorithms [34].

2.2. Musically-Structured dictionary

The dictionary matrix \mathbf{A} can be chosen in three different manners:

- (1) Analytic dictionary – a predefined dictionary, which is the inverse (or pseudo-inverse) matrix of a transform such as the Curvelet transform [38,39], the Contourlet transform [40,41] the short-time Fourier transform or the Wavelets transform [42].
- (2) Learned (explicit) dictionary – a dictionary is learned blindly from a set of signals or samples $\mathbf{y}_i, (1 \leq i \leq M)$, in an iterative and alternative manner, arranged as columns of a matrix \mathbf{Y} . The optimization problem (P_0^ε) turns into:

$$\min_{\mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 < K, \quad 1 \leq i \leq M. \quad (3)$$

Abdallah and Plumbley [22,24] used such a learned dictionary for transcription of polyphonic music.

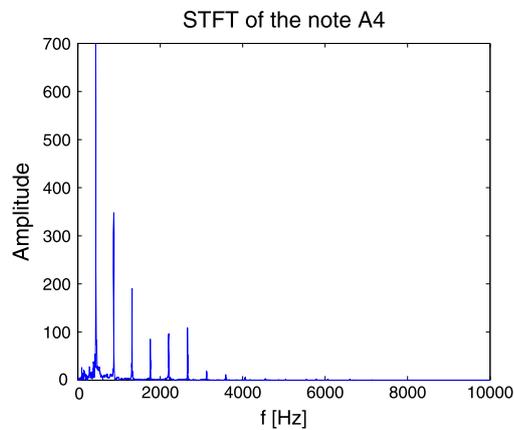


Fig. 1. Short-time Fourier transform of an A4 piano note.

- (3) Parametric dictionary – a compromise between a pre-defined analytic dictionary to a data-driven learned dictionary. Only a limited set of parameters in the dictionary is learned, and the rest is pre-defined. It represents the data better than an analytic dictionary, and is computationally more efficient than an explicit learned dictionary. In addition, it can avoid overfitting if built wisely, unlike an explicit learned dictionary.

Our motivation is to develop a parametric dictionary which is suitable for the spectrum of music signals. In order to do so, we use the common features in the spectrum of a musical note. The magnitude of the Fourier transform of every note can be modeled by an impulse train at the fundamental frequency and at its harmonics, which has been multiplied with a shaping filter, that is controlled by the factors mentioned in Section 1 – the musical instrument, the intensity, the duration of the note and the fundamental frequency. For illustration, see the short-time Fourier transform (STFT) of an A4 piano note in Fig. 1. This STFT illustration was performed on a note played over a long time, so we could use a large time window, and get good resolution both in time and in frequency.

If we apply the Constant Q transform instead of the STFT on a A4 piano note, we get peaks of the fundamental frequency and its harmonics in intervals which become logarithmically smaller as the CQT bin grows (see bottom of Fig. 2). The dictionary which we offer is initialized such that each atom is the modeled CQT of a different note in the piano (total – 88 notes). More specifically, we initialize each atom by an impulse train of 6 elements, corresponding to the suitable fundamental frequency and its first 5 harmonics. This number of harmonics was found to be sufficient in multiple tests and on a wide range of notes. We multiply this finite impulse train by an initial shaping function $f(n) = \frac{1}{n}$, where n is the partial number. Finally, we map the dictionary to the CQT scale. We denote this dictionary as the initial *Musically-Structured dictionary*. An image of this initial dictionary, and the CQT of a note represented by a certain atom in it, are presented in Fig. 2.

The next step is learning certain parameters in the Musically-Structured dictionary. The *support* of the dictionary is constant, i.e., the location of the non-zero elements in the vector, which are the CQT bins of the fundamental frequencies and their 5 first harmonics. However, the entries of the elements in the support are learned, i.e., the amplitudes of the fundamental frequencies and their harmonics. These amplitudes are an important feature of the *timbre* of the signal, which was mentioned before. The minimal resolution required for music transcription is 12 CQT frequency bins per octave (one for each 100 cent = semitone). However, using a higher resolution improves the transcription results for poly-

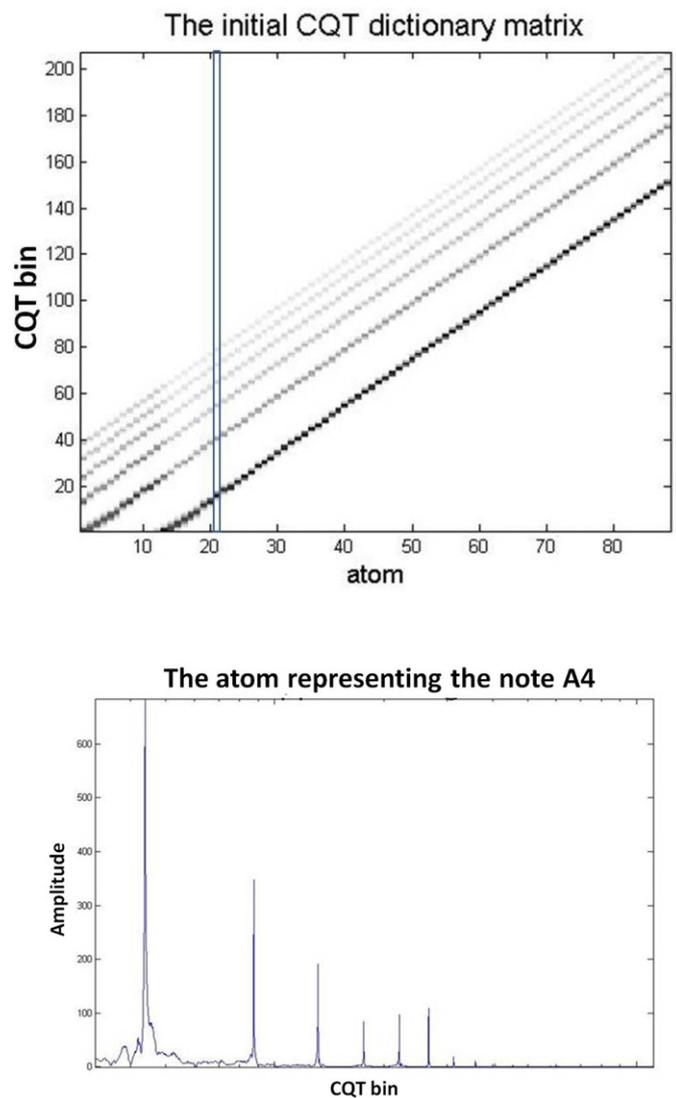


Fig. 2. The initial MS dictionary (top) and the CQT of an A4 note represented by a certain atom (bottom). The atom which represents the note in the bottom picture is marked by a rectangle in the dictionary.

phonic music. In the experiments presented in Section 4 we show the results for a mapping to 24 frequency bins per octave (one for each 50 cents = quarter tone).

The MS-dictionary is expected to have several advantages over the explicit learned dictionary proposed by Abdallah and Plumbley [24], as well as advantages over an analytic and pre-defined appropriate dictionary:

- (1) Avoids overfitting – each atom in the MS dictionary represents a single note, even if it does not appear individually in the dataset (e.g., it is played only as part of a chord). We expect this requirement to be fulfilled, since the support of the dictionary is constant, and since each atom in the initial dictionary represents an individual note.
- (2) Better representation of the signal than an analytic dictionary – this is due to the fact that the entries in the support of the dictionary matrix are learned according to the timbre of the signal.
- (3) Reduced complexity – the complexity of learning the dictionary is proportional to the number of the learned parameters. In an explicit dictionary, the number of learned parameters equals the size of the dictionary, i.e., nm . In the MS-dictionary,

the number of learned parameters is hm , where h is the number of partials (fundamental frequency + harmonics), which we choose to be 6. Since $h < n$, the complexity of the dictionary is reduced.

The formulation of the new problem is:

$$(P_{MS}): \min_{\mathbf{A}, \mathbf{X}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq K \quad \forall i \in \{1, \dots, m\}$$

$$\mathbf{a}_j(P_j^c) = 0 \quad \forall j \in \{1, \dots, M\}, \quad (4)$$

where P_j^c is the subset of indices out of the support of the atom \mathbf{a}_j . In order to approximate the solution to P_{MS} , we apply a modified MOD (Method-Of-Directions) or K-SVD dictionary learning algorithm according to the MS parametric dictionary.

The MS-dictionary is composed of 88 atoms, where each atom represents a different note. Musical pieces usually do not contain all of the 88 notes. Therefore, in the dictionary learning stage, we update only the atoms which were used for the sparse coding stage. We denote the dictionary matrix of the used atoms in the k th iteration as $\tilde{\mathbf{A}}^{(k)}$, and the corresponding coefficients matrix as $\tilde{\mathbf{X}}^{(k)}$. The rest of the atoms remain unchanged, and are added to the dictionary after the update. In the following, we describe the modified MOD and K-SVD dictionary learning algorithms.

2.3. Musically-Structured MOD

In the MOD algorithm, developed by Engan et al. [43], the dictionary update stage is conducted using least squares:

$$\mathbf{A}^{(k)} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}^{(k)}\|_F^2 = \mathbf{Y}\mathbf{X}^{(k)\dagger}, \quad (5)$$

where \mathbf{X}^\dagger is the Moore–Penrose pseudo-inverse of \mathbf{X} . After an initialization of the dictionary matrix, the matrix of sparse columns \mathbf{X} and the dictionary matrix \mathbf{A} are updated alternately at each iteration, until the change at the k th iteration of $\|\mathbf{Y} - \mathbf{A}^{(k)}\mathbf{X}^{(k)}\|_F^2$ is small enough.

In our modified MOD algorithm, the MS-MOD, the dictionary matrix is updated as in the MOD algorithm:

$$\tilde{\mathbf{A}}^{(k)} = \underset{\tilde{\mathbf{A}}}{\operatorname{argmin}} \|\tilde{\mathbf{Y}} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}^{(k)}\|_F^2 = \tilde{\mathbf{Y}}\tilde{\mathbf{X}}^{(k)\dagger}. \quad (6)$$

The matrix $\tilde{\mathbf{X}}$ is composed of the concatenation of the rows of \mathbf{X} which have a ℓ_1 -norm above a certain threshold, and correspond to the used atoms in \mathbf{A} . This diminution is intended to prevent singularity of \mathbf{X}^\dagger , and deals with (most of) the cases, where the musical piece does not contain all of the 88 notes.

The normalization stage of the atoms in the MOD (and in the K-SVD) is intended to make the implementation of the sparse coding stage simpler when using a greedy method, and it does not change its solution.

After updating the atoms, we zero the elements out of the original support. The MS-MOD is summarized in Table 1. The main modifications compared to the original MOD are indicated by underlined text.

2.4. Musically-Structured K-SVD

A different update rule for the dictionary was proposed by Aharon et al. [44], leading to the K-SVD algorithm. In this algorithm, the atoms (i.e., columns) in the dictionary \mathbf{A} are handled sequentially. The dependency on the atom \mathbf{a}_{j_0} in (3) is isolated by rewriting the term $\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2$:

Table 1

The Musically-Structured MOD algorithm.

Task: Train a dictionary \mathbf{A} to sparsely represent the log spectrogram matrix \mathbf{Y} , by approximating the solution to Problem (4).

Initialization: Initialize $k = 0$, and

- **Initialize Dictionary:** Build $\mathbf{A}_{(0)} \in \mathbb{R}^{n \times m}$, with non-zero entries only in the locations corresponding to the fundamental frequencies and their harmonics.
- **CQT mapping:** Map the dictionary to the CQT logarithmic scale.
- **Normalization:** Normalize the columns of $\mathbf{A}_{(0)}$ to unit ℓ_2 -norm.

Main iteration: Increment k by 1, and apply

- **Sparse Coding Stage:** Use a pursuit algorithm to approximate the solution of $\hat{\mathbf{x}}_i = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{A}_{(k-1)}\mathbf{x}\|_2^2$ subject to $\|\mathbf{x}\|_0 \leq K$ obtaining sparse representations $\hat{\mathbf{x}}_i$ for $1 \leq i \leq M$. These form the matrix $\mathbf{X}_{(k)}$.
- **Define the used atoms:** The rows in $\mathbf{X}_{(k)}$ which have an ℓ_1 -norm above a certain threshold correspond to the used atoms in the dictionary matrix. These rows are concatenated and defined as $\tilde{\mathbf{X}}^{(k)}$ and the matrix composed of the used atoms is defined as $\tilde{\mathbf{A}}$.
- **MOD Dictionary Update Stage:** Update the dictionary of used atoms $\tilde{\mathbf{A}}^{(k)}$ by the matrix $\tilde{\mathbf{X}}^{(k)}$: $\tilde{\mathbf{A}}^{(k)} = \operatorname{argmin}_{\tilde{\mathbf{A}}} \|\mathbf{Y} - \tilde{\mathbf{A}}\tilde{\mathbf{X}}^{(k)}\|_F^2 = \mathbf{Y}\tilde{\mathbf{X}}^{(k)\dagger}$.
- **Zero the entries out of the support:** Zero the entries out of the support which was defined in the initial dictionary.
- **Add the unused atoms:** Add the unused atoms to $\tilde{\mathbf{A}}^{(k)}$. This is the updated dictionary $\mathbf{A}^{(k)}$.
- **Stopping Rule:** If the change in $\|\mathbf{Y} - \mathbf{A}^{(k)}\mathbf{X}^{(k)}\|_F^2$ is small enough, stop. Otherwise, apply another iteration.

Output: The desired outputs are $\mathbf{A}^{(k)}$ and $\mathbf{X}^{(k)}$.

$$\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 = \left\| \mathbf{Y} - \sum_{j=1}^m \mathbf{a}_j \mathbf{x}_j^T \right\|_F^2$$

$$= \left\| \left(\mathbf{Y} - \sum_{j \neq j_0} \mathbf{a}_j \mathbf{x}_j^T \right) - \mathbf{a}_{j_0} \mathbf{x}_{j_0}^T \right\|_F^2. \quad (7)$$

In this description \mathbf{x}_j^T stands for the j th row of \mathbf{X} , i.e., the coefficients which correspond to the j th atom. We define the term

$$\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j \neq j_0} \mathbf{a}_j \mathbf{x}_j^T \quad (8)$$

as the error matrix corresponding to the atom \mathbf{a}_{j_0} . We restrict it only to the columns that correspond to the signals (columns) in \mathbf{Y} which use the atom \mathbf{a}_{j_0} , and denote the restricted error matrix as $\mathbf{E}_{j_0}^R$. Both \mathbf{a}_{j_0} and the non-zero elements in $\mathbf{x}_{j_0}^T$, which are denoted by $(\mathbf{x}_{j_0}^R)^T$, are updated in this algorithm, by minimizing the term in (7), using a rank-1 approximation of the error matrix $\mathbf{E}_{j_0}^R$. This approximation is obtained via singular value decomposition (SVD).

In our modified algorithm, the MS-KSVD, we update only the elements in the support, for each atom individually. For each atom \mathbf{a}_{j_0} , the error matrix \mathbf{E}_{j_0} is defined as in the K-SVD. Its columns are restricted as in the K-SVD, but now its rows are also restricted, according to the support of the atom \mathbf{a}_{j_0} . Thus, updating only the support of \mathbf{a}_{j_0} , we denote this restricted error matrix as $\tilde{\mathbf{E}}_{j_0}^R$ and the elements in the support of \mathbf{a}_{j_0} as $\tilde{\mathbf{a}}_{j_0}$. The vectors $\tilde{\mathbf{a}}_{j_0}$ and $(\mathbf{x}_{j_0}^R)^T$ are updated using rank-1 approximation of the error matrix $\tilde{\mathbf{E}}_{j_0}^R$ by singular value decomposition. The algorithm is summarized in Table 2. The main modifications compared to the original K-SVD are indicated by underlined text.

3. Implementation

The overall transcription algorithm which we offer is as follows:

- (1) Note onset detection – since we focus only on pitch identification, we conduct this stage manually, or extract the onsets

Table 2
The Musically-Structured K-SVD algorithm.

Task: Train a dictionary \mathbf{A} to sparsely represent the log spectrogram matrix \mathbf{Y} , by approximating the solution to Problem (4).

Initialization: Initialize $k = 0$, and

- **Initialize Dictionary:** Build $\mathbf{A}_{(0)} \in \mathbb{R}^{n \times m}$, with non-zero entries only in the locations corresponding to the fundamental frequencies and their harmonics.
- **CQT mapping:** Map the dictionary to a the CQT logarithmic scale.
- **Normalization:** Normalize the columns of $\mathbf{A}_{(0)}$ to unit ℓ_2 -norm.

Main iteration: Increment k by 1, and apply

- **Sparse Coding Stage:** Use a pursuit algorithm to approximate the solution of $\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{A}_{(k-1)}\mathbf{x}\|_2^2$ subject to $\|\mathbf{x}\|_0 \leq K$ obtaining sparse representations $\hat{\mathbf{x}}_i$ for $1 \leq i \leq M$. These form the matrix $\mathbf{X}_{(k)}$.
- **KSVD Dictionary Update Stage:** Update the support of each atom $\tilde{\mathbf{a}}_{j_0, j_0=1, 2, \dots, m}$ in the dictionary matrix by rank-1 approximation of its error matrix $\tilde{\mathbf{E}}_{j_0}^R$, using SVD. $\tilde{\mathbf{E}}_{j_0}^R$ is the restriction of the matrix $\mathbf{E}_{j_0} = \mathbf{Y} - \sum_{j \neq j_0} \mathbf{a}_j \mathbf{x}_j^T$ to the columns that correspond to the samples in \mathbf{Y} which use the atom \mathbf{a}_{j_0} , and to the rows that correspond to the support of \mathbf{a}_{j_0} . After applying the SVD $\tilde{\mathbf{E}}_{j_0}^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$, update the support of the dictionary atom by $\tilde{\mathbf{a}}_{j_0} = \mathbf{u}_1$ and their representation coefficients by $(\mathbf{x}_{j_0}^R)^T = (\mathbf{\Delta}[1, 1]\mathbf{v}_1)^T$.
- **Add the unused atoms:** Add the unused atoms to $\tilde{\mathbf{A}}_{(k)}$. This is the updated dictionary $\mathbf{A}_{(k)}$.
- **Stopping Rule:** If the change in $\|\mathbf{Y} - \mathbf{A}_{(k)}\mathbf{X}_{(k)}\|_F^2$ is small enough, stop. Otherwise, apply another iteration.

Output: The desired outputs are $\mathbf{A}_{(k)}$ and $\mathbf{X}_{(k)}$.

from a MIDI file in the case of transcribing a synthesized MIDI musical piece.

- (2) Evaluation of the number of notes – performed 32 ms after the onset, in a 64 ms time window, either manually or extracted from a MIDI file (from the same reason as in the previous stage). This number is defined as K , it is given as input to our algorithm at each relevant time frame, and it is used as the maximal cardinality of the sparse vector during the transcription process.
- (3) Constant Q transform is applied on the signal in the 64 ms time window.
- (4) All the vectors of CQTs of the time windows mentioned before are concatenated in columns of the signal matrix \mathbf{Y} (which is actually a log-spectrogram), and a Musically-Structured dictionary learning algorithm is applied on the matrix \mathbf{Y} to transcribe the music in each of the time windows represented by its columns.

The reason for applying the multi-pitch estimation only on 64 ms time windows, 32 ms after the onsets of the notes, is that the acoustic spectrum of a tone significantly changes as a function of the stage in the ADSR envelope. We wish to sample all notes at the same stage, such that the atoms in the dictionary would represent them well. We assume that after 32 ms, the ADSR envelope is in its *sustained* stage, which is the most suitable stage for the spectral analysis because of its stability and relatively long duration. A block diagram of the overall algorithm is presented in Fig. 3.

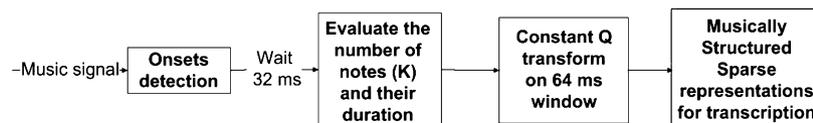


Fig. 3. A block diagram of the overall transcription algorithm.

4. Experiments

4.1. Experimental setup

We perform pitch identification of synthesized piano music from MIDI files, and pitch identification of real recorded piano music. We do not deal with finding the onset and offset instances of the notes, neither with finding the number of notes played at each time (which we use as the maximal number of notes – K , for the sparse coding). The stopping criterion for the algorithms is achieved when there is a change of less than 5% in the Frobenius norm of the residual. We compare the performance of the MS-MOD and MS-K-SVD to that of the standard MOD and K-SVD algorithms (which use an explicit, blindly learned dictionary), and to that of an analytic dictionary (un-learned) with OMP in the sparse coding stage. The analytic dictionary is determined as the initial MS-dictionary. In the standard MOD and K-SVD we decide on the fundamental frequency of individual atoms that are learned, by a method called HPS (Harmonic Product Spectrum), [45] which is intended for pitch estimation in monophonic music.

We use different measures for the evaluation of the results: The first is the *Accuracy* measure, defined by [46]

$$Accuracy = \frac{TP}{TP + FP + FN}. \quad (9)$$

The term TP is the number of true positives (correct detections), FP is the number of false positives and FN is the number of false negatives. When $Accuracy = 1$, it means that all the notes are identified correctly and there are no false positives nor false negatives. The second measure is the *transcription error score* [12]. If we denote by N_{sys} the number of reported pitches, by N_{ref} the number of ground-truth pitches and by N_{corr} their intersection, then the transcription error score across all time frames t is:

$$E_{\text{tot}} = \frac{\sum_{t=1}^T \max(N_{\text{ref}}(t), N_{\text{sys}}(t)) - N_{\text{corr}}(t)}{\sum_{t=1}^T N_{\text{ref}}(t)}.$$

The MIDI files for the experiments include a monophonic musical piece, a simple polyphonic piece (mainly two or three notes played at a time, a limited variety of notes), a complicated polyphonic piece (mainly five or six notes played at a time, a large variety of notes) and a piece of chords. Their piano rolls are presented in the next subsection. These MIDI files are synthesized with a sampling frequency of 44.1 kHz, by FM-synthesis, using the “Matlab and MIDI” software [47]. The MS-K-SVD code is a modified version of the K-SVD code [48]. The recorded piano files include a monophonic piece and a piece of chords. These pieces are recorded on a Yamaha U1 piano, and saved with a sampling frequency of 44.1 kHz.

We also compare our transcription method to former reported transcription results (Costantini et al. [15], Poliner and Ellis [12], Rynnänen and Klapuri [21] and Marolt [10]), which were examined on a set of polyphonic classical synthesized MIDI music pieces which were collected from the Classical Piano Midi Page [49]. The list of 130 pieces set is specified in [12]. The first minute from each song was taken. The 130 pieces set was randomly split into 92 training, 24 testing and 13 validation pieces (we used for our tests only the testing set since no training is needed). In addition to the

synthesized audio, piano recordings were made from a subset of the classical MIDI files using a Yamaha Disklavier playback grand piano. Twenty training files and ten testing files were randomly selected for recording. The recorded files are available at [50].

4.2. Results – synthesized MIDI music

First we present the performance of the MS-dictionary learning algorithm on a monophonic piece, a simple polyphonic piece and a complicated polyphonic piece. The transcription results are presented in Tables 3 and 4. The corresponding piano rolls and their identification by MS-MOD and MS-KSVD are presented in Figs. 4, 5 and 6. From the tables we can learn that:

- (1) All methods perfectly identify the notes in the monophonic music.

Table 3
Transcription Accuracy percentage for three different types of songs.

	Monophonic music	Simple polyphonic music	Complicated polyphonic music
MS-MOD	100	69.6	64.0
MS-K-SVD	100	67.7	64.5
MOD	100	39.5	43.5
K-SVD	100	37.6	42.7
Analytic dictionary	100	45.8	41.0

Table 4
Transcription E_{tot} percentage for three different types of songs.

	Monophonic music	Simple polyphonic music	Complicated polyphonic music
MS-MOD	0	17.9	23.6
MS-K-SVD	0	19.3	23.2
MOD	0	44.5	41.0
K-SVD	0	46.2	41.7
Analytic dictionary	0	37.2	43.3

- (2) The estimation performance of the algorithms based on the parametric MS dictionary is better than that of the explicit dictionaries as well as the analytic dictionaries when identifying notes in polyphonic music. The advantage is more significant in the simple polyphonic music than in the complicated polyphonic music (where the dataset of notes is larger and richer).
- (3) The estimation performance of the MS-dictionary and the analytic dictionary become worse when the polyphonic music becomes more complicated. On the contrary, the performances of the methods based on explicit dictionaries are improved. This implies that the explicit dictionaries need a large data-set in order to achieve good performance.

From the piano rolls we can identify some mistakes in the transcription as spurious notes (mistakes of semitones), which might be fixed by using a higher frequency resolution. Some mistakes are associated with notes that share a similar spectral shape, such as notes whose difference is an integer number of octaves, or notes that share common harmonics with the true note.

We now turn to the problem mentioned in Section 1 – deduction of notes from chords. This problem is hard, since the notes in a chord share multiple harmonics. We compare the results of the MS-dictionary to that of an explicit dictionary, and of an analytic dictionary. The piano rolls of the original and identified music are presented in Fig. 7. In the case of the chords, MS-MOD, MS-K-SVD and OMP with an analytic dictionary identify all the notes, despite of the difficulty of this task. The MOD and K-SVD identify only the lower notes.

We compare the former reported results, obtained by the implementations of original authors (Costantini et al. [15], Poliner and Ellis [12], Ryyänen and Klapuri [21] and Marolt [10]), on the set of polyphonic classical music, which was described in the beginning of this section, to the multi-pitch estimation of the testing set by MS-MOD, MS-K-SVD, MOD, K-SVD and OMP with an ana-

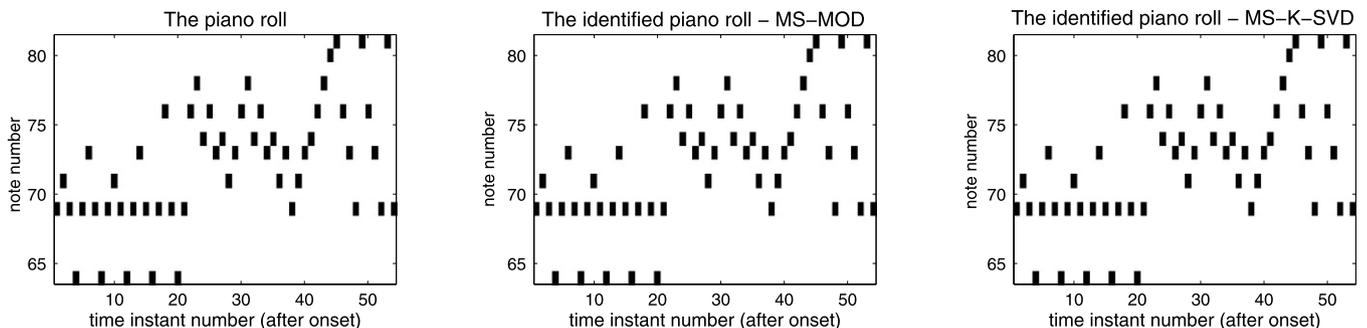


Fig. 4. The ground-truth piano roll (left) and the identified piano roll of a monophonic piece using MS-MOD (middle) and using MS-K-SVD (right). All the notes were identified correctly.

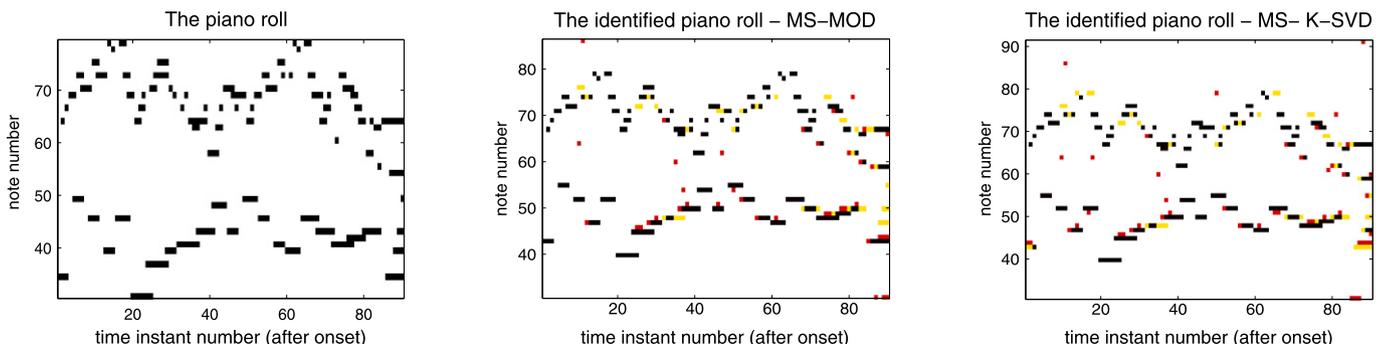


Fig. 5. The ground-truth piano roll (left) and the identified piano roll of a simple polyphonic piece using MS-MOD (middle) and using MS-K-SVD (right). Black = true positive, red = false positive, yellow = false negative. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

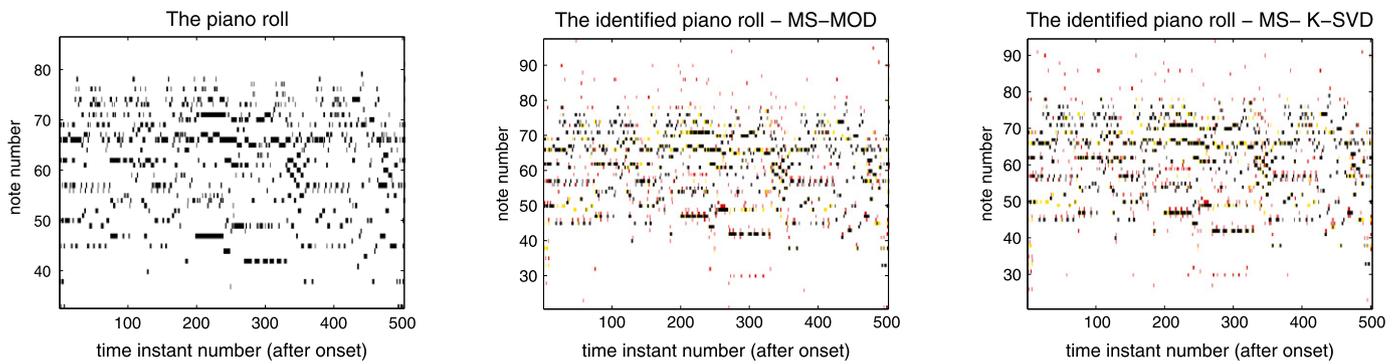


Fig. 6. The ground-truth piano roll (left) and the identified piano roll of a complicated polyphonic piece using MS-MOD (middle) and using MS-K-SVD (right). Black = true positive, red = false positive, yellow = false negative. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

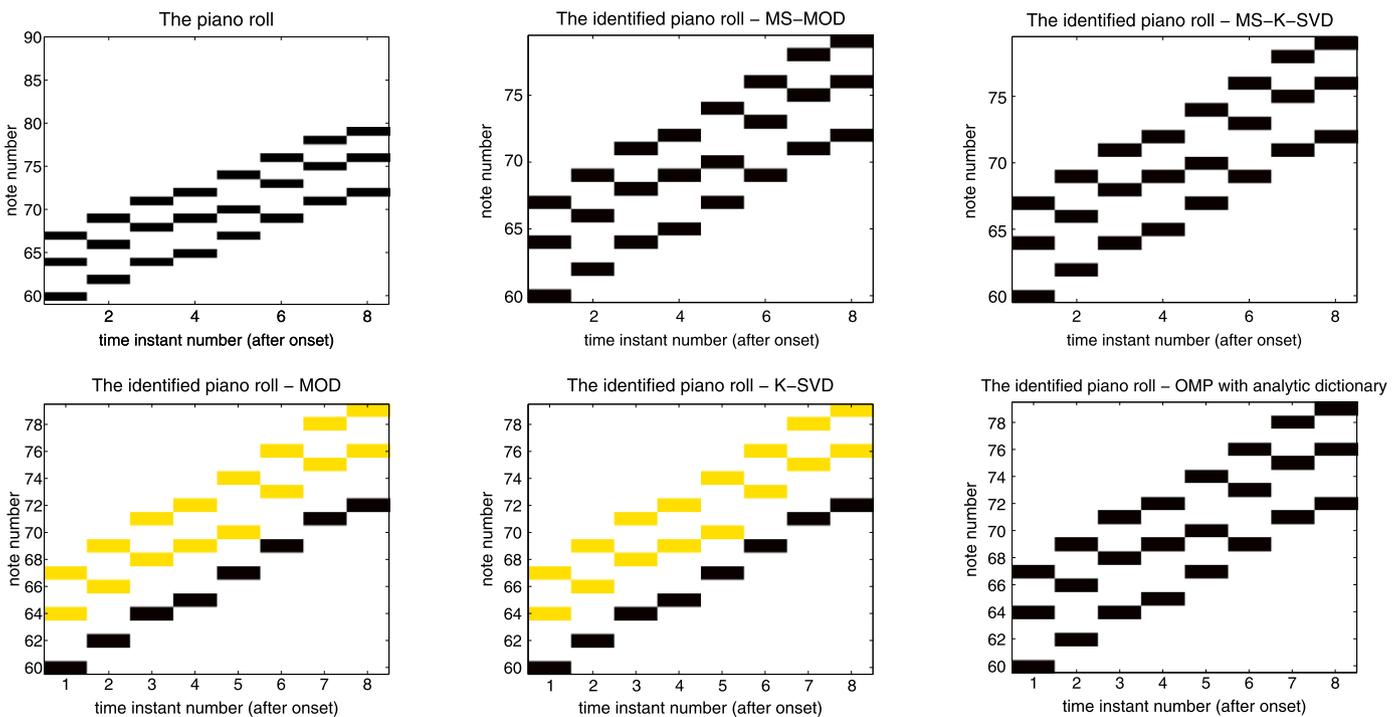


Fig. 7. The ground-truth piano roll and the identified piano roll of synthesized chords using different methods. Black = true positive, red = false positive, yellow = false negative. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

lytic dictionary. However, the results should be compared carefully due to the following differences between the algorithms:

- (1) The former transcription algorithms are supervised methods, i.e. they are based on the training set mentioned before. Their results presented here are after training on pieces which were written by the same composers as in the testing set. Our algorithm, and the other transcription methods based on sparse coding, are unsupervised methods, and they are tested on the same testing set *without training*.
- (2) The number of notes in each frame was inserted as a parameter to the pitch estimation methods based on sparse representations (MS-MOD, MS-K-SVD, MOD, K-SVD and OMP with an analytic dictionary), as the maximal cardinality K of each vector of coefficients \mathbf{x} . However, the former transcription algorithms did not receive the maximal cardinality as a parameter.

A comparison of the results on synthesized polyphonic classical music are presented in Table 5. From the table one can see

Table 5

Transcription performance on synthesized polyphonic classical music.

	Accuracy (%)	E_{tot} (%)
MS-MOD	58.7	28.2
MS-K-SVD	59.8	27.2
MOD	39.4	45.3
K-SVD	31.0	54.1
Analytic dictionary	39.1	45.1
Costantini et al.	72.3	20.1
Poliner and Ellis	64.7	41.7

that the results of the transcription by MS-dictionary learning algorithms outperform those of the other unsupervised methods for transcription using sparse representations (MOD, K-SVD and OMP with an analytic dictionary). They are inferior compared to those of other transcription methods, but, as mentioned before, not directly comparable. The measures *Accuracy* and E_{tot} are not always correlated in their performance, due to their different definitions. Therefore, they are both used for evaluation.

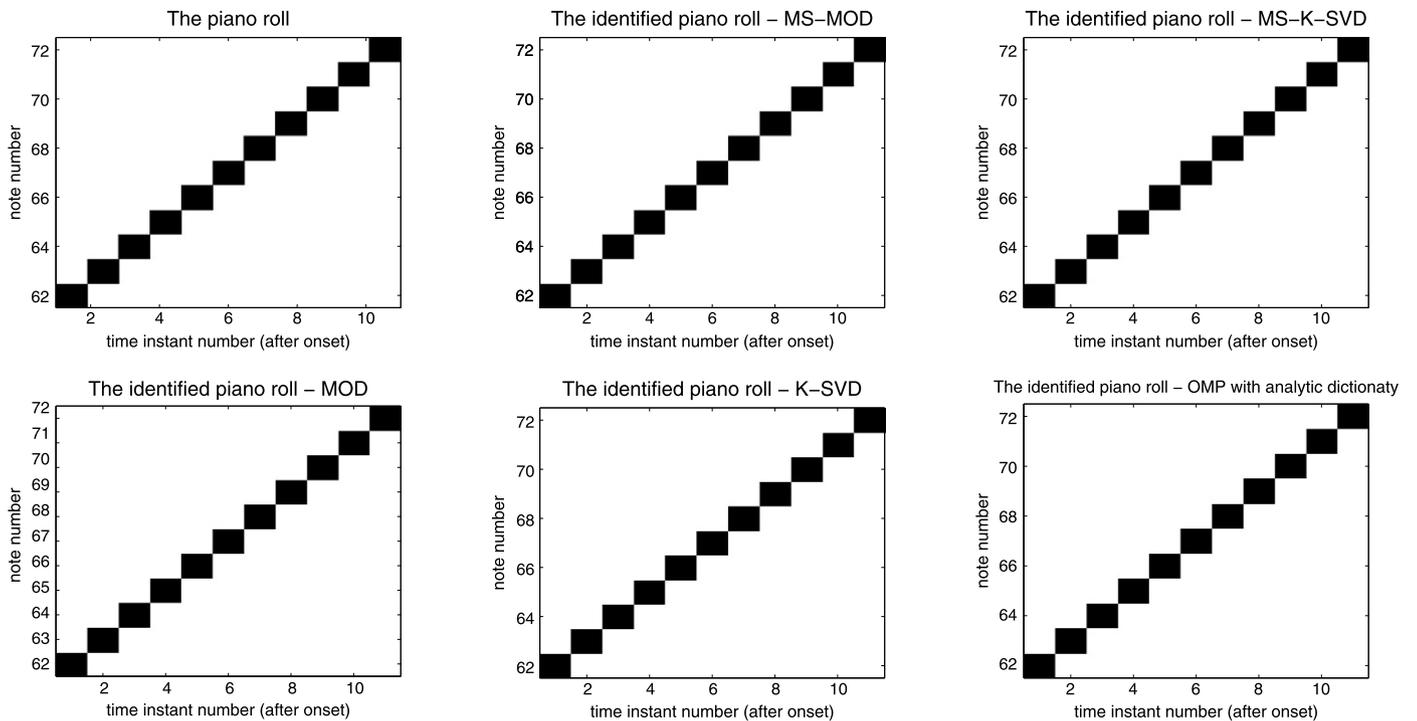


Fig. 8. The ground-truth piano roll and the identified piano roll of recorded monophonic piano music using different transcription methods. All the notes were identified correctly.

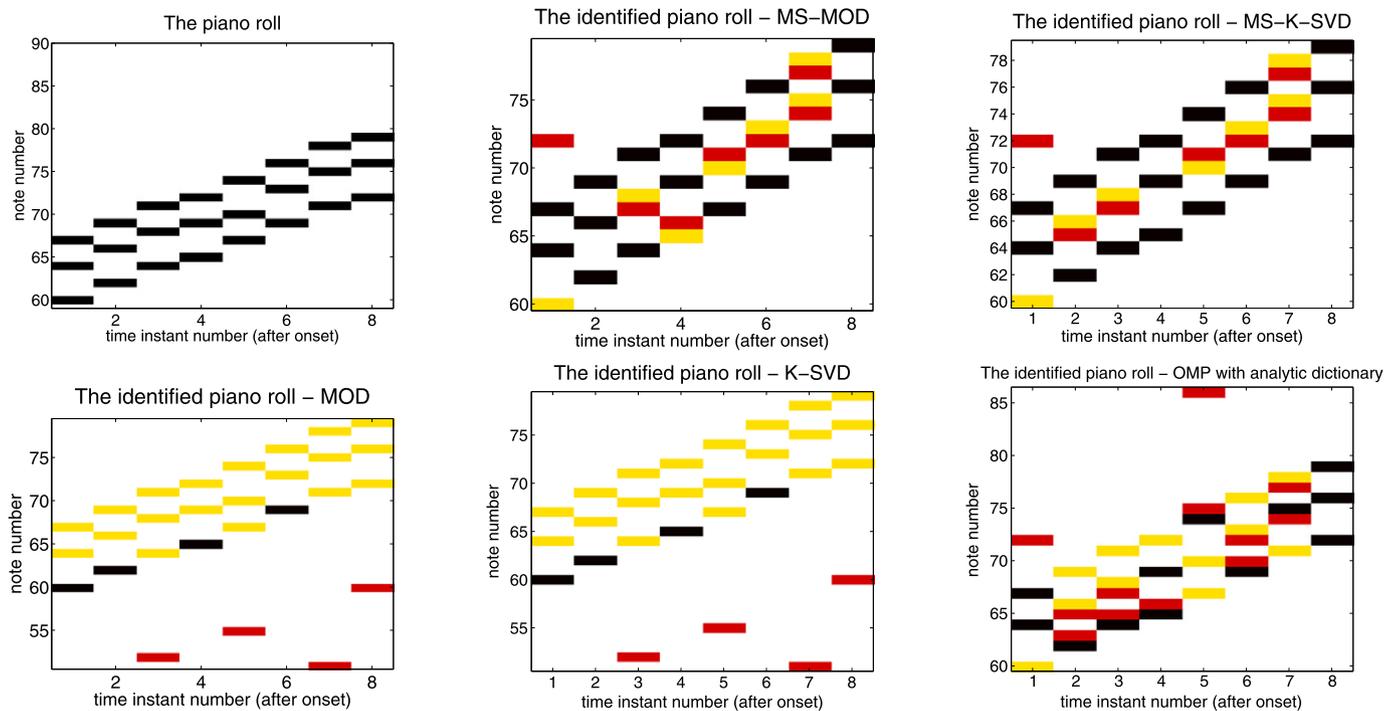


Fig. 9. The ground-truth piano roll and the identified piano roll of recorded piano chords using different transcription methods. Black = true positive, red = false positive, yellow = false negative. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Results – recorded piano music

The task of multiple pitch estimation of real recorded piano music is much harder, since it entails several additional obstacles, such as beginning transients of notes, a noisy environment, formants produced by the shape of the musical instrument, reverberations of the room and strings inharmonicity. In a real vibrating

string, the harmonics are not located exactly in multiple integers of the fundamental frequency, and the musical instruments are tuned accordingly [51].

First we perform some simple tests, on monophonic music and on chords. We compare the performances of MS-MOD and MS-K-SVD to that of MOD, K-SVD, and OMP with an analytic dictionary, as presented in Figs. 8 and 9. All the methods perfectly iden-

Table 6
Transcription performance on recorded polyphonic classical music.

	Accuracy (%)	E_{tot} (%)
MS-MOD	54.3	30.9
MS-K-SVD	54.3	31.1
MOD	22.9	63.3
K-SVD	19.8	67.4
Analytic dictionary	43.9	40.1
Costantini et al.	59.2	33.3
Poliner and Ellis	56.5	46.7

Table 7
Transcription performance on synthesized plus recorded polyphonic classical music.

	Accuracy (%)	E_{tot} (%)
MS-MOD	57.4	29.0
MS-K-SVD	58.2	28.3
MOD	34.5	50.6
K-SVD	27.7	58.0
Analytic dictionary	40.5	43.6
Costantini et al.	68.0	24.6
Poliner and Ellis	62.3	43.2
Ryynänen and Klapuri	56.8	46.0
Marolt	30.4	87.5

tify the notes in the monophonic music. The detection of chords is worse than in synthesized music, and this may stem from the obstacles mentioned before. Still, the performances of the MS-MOD, MS-K-SVD and OMP with an analytic dictionary are better than those of the MOD and K-SVD, which identify one note at a time while there are actually two or three notes.

We compare the transcription results on the recorded classical polyphonic music pieces to those of previous works. The results of transcription on recorded polyphonic music are presented in Table 6, and the results of transcription on both synthesized and recorded polyphonic music are presented in Table 7. In recorded classical piano music, similarly to synthesized classical piano music, the results of the transcription by MS-dictionary learning algorithms outperform those of the other unsupervised methods for transcription using sparse representations (MOD, K-SVD and OMP with an analytic dictionary). In this case they have similar results to other transcription methods, and even outperform some of them, although they are supervised methods.

5. Conclusions

We have presented a multiple fundamental frequency estimation system based on sparse representations. The power spectrum (CQT) of the music signal at each time window is represented as a multiplication of a learned dictionary with a sparse vector. This framework relies on the assumptions that the number of notes played at a time is small compared to the number of available notes, and that the power spectrum of different notes approximately add, for random phase relationships. We offered a parametric dictionary, namely “Musically-Structured dictionary”, based on the common features of the power spectra of music signals. This parametric dictionary is more suitable for multiple pitch estimation than an analytic dictionary or an explicit dictionary. We developed modifications of two dictionary learning algorithms, MOD and K-SVD, which are denoted MS-MOD and MS-K-SVD, respectively, for learning the parametric dictionary.

The performance of the MS-dictionary was examined empirically on MIDI synthesized music and on recorded music. In the polyphonic pieces we examined, most of the notes were recognized correctly. Relatively good performance was seen also in cases of small datasets with overlapping harmonics, such as chords or octaves. It is shown that the transcription using MS-dictionary outperforms transcription using an analytic or an explicit dictionary.

The advantage over an explicit dictionary grows as the dataset is smaller, and as there are more overlapping harmonics.

The proposed algorithm is advantageous from several reasons: It adapts to the timbre of the signal and reduces computational complexity compared to sparse representations with analytic dictionaries. It avoids overfitting to small datasets or to notes played conjugately (such as chords), thus outperforming sparse representations with explicit dictionaries. In addition, it is an unsupervised method, thus a training set is not required. Our framework can be modified and expanded in several ways: The dictionary can be better adapted to real recorded music. The overtones of real strings are not exactly multiple integers of the fundamental frequency (string inharmonicity). Therefore, a dictionary in which the non-zero elements are non-uniformly spaced, might lead to better transcription results. Another possible expansion is to add more atoms to the dictionary. The timbre of the sound changes as a function of its intensity. Therefore, it is reasonable that each note would be represented by several atoms, where each atom represents a different intensity. Each group of atoms which represents the same note would have the same support, but the values of the non-zero elements would be changed individually, such that each atom would fit another time window (or group of time windows) within the note.

Another possibility is to exploit high-level information. Former works [9,12] used prior information on the structure, tempo, rhythm and key of the musical piece, as well as expected relations between consecutive notes. This can be added to our framework as a subsequent stage, or as an iterative stage, for example via a Hidden Markov Model (HMM).

As for the comparison to non-negative matrix factorization (NMF) mentioned in Section 1, we would like to point out the similarities and dissimilarities to our work. Both approaches – NMF and sparse representations, deal with the factorization problem $\mathbf{AX} = \mathbf{Y}$. Moreover, sparse representations can be considered as a special case of NMF, in which the sparseness constraint is imposed [34]. In addition, in the specific NMF algorithms mentioned in Section 1, the initialization and update of the matrix \mathbf{A} are based on a special “musically-based” structure, similar to ours. However, the major difference between the NMF approach and our approach is in the update rule – while the NMF algorithms update both matrices \mathbf{A} and \mathbf{X} by minimizing the Bregman divergence between the matrix \mathbf{Y} and its approximation \mathbf{AX} , we use pursuit algorithms for estimating the matrix \mathbf{X} , and dictionary learning algorithms for estimating the matrix \mathbf{A} . These algorithms are addressed specifically for approximating the sparse representation of the signal, with our adaptation to the problem of musical multi-pitch estimation. An interesting test would be to compare the multi-pitch estimation of our algorithm to that of the relevant NMF algorithms, on the same data set.

A next step would be to extend the proposed framework to multi-pitch estimation for other musical instruments, as well as for their combinations. It would be interesting to examine the difference in the performance of the algorithm between western and non-western instruments (e.g., [52]).

Acknowledgments

The authors thank Dr. Ron Rubinstein for fruitful discussion and helpful suggestions. They also thank the anonymous reviewers for their constructive comments and useful suggestions.

References

- [1] K. Martin, A blackboard system for automatic transcription of simple polyphonic music, Massachusetts Institute of Technology Media Laboratory Perceptual Computing Section, Tech. Rep., 1996.

- [2] A. Klapuri, Automatic transcription of music, Master's thesis, Tampere University of Technology, Tampere, Finland, 1998.
- [3] A. Klapuri, Automatic music transcription as we know it today, *J. New Music Res.* 33 (3) (2004) 269–282.
- [4] J. Moorer, On the segmentation and analysis of continuous musical sound by digital computer, Ph.D. dissertation, 1975.
- [5] R. Meddis, M. Hewitt, Virtual pitch and phase sensitivity of a computer model of the auditory periphery. I. Pitch identification, *J. Acoust. Soc. Am.* 89 (6) (1991) 2866–2882.
- [6] R. Meddis, L. O'Mard, A unitary model of pitch perception, *J. Acoust. Soc. Am.* 102 (1997) 1811–1820.
- [7] A. de Cheveigné, H. Kawahara, YIN, a fundamental frequency estimator for speech and music, *J. Acoust. Soc. Am.* 111 (2002) 1917–1930.
- [8] M. Plumbley, S. Abdallah, T. Blumensath, M. Davies, Sparse representations of polyphonic music, *Signal Process.* 86 (3) (March 2006) 417–431.
- [9] A. Cemgil, H. Kappen, D. Barber, A generative model for music transcription, *IEEE Trans. Audio, Speech Language Process.* 14 (2) (March 2006) 679–694.
- [10] M. Marolt, A connectionist approach to automatic transcription of polyphonic piano music, *IEEE Trans. Multimedia* 6 (3) (June 2004) 439–449.
- [11] A. Klapuri, Multipitch analysis of polyphonic music and speech signals using an auditory model, *IEEE Trans. Audio, Speech Language Process.* 16 (2) (2008) 255–266.
- [12] G. Poliner, D. Ellis, A discriminative model for polyphonic piano transcription, *EURASIP J. Appl. Signal Process.* 2007 (1) (January 2007) 154–162.
- [13] J. Brown, Musical fundamental frequency tracking using a pattern recognition method, *J. Acoust. Soc. Am.* 92 (3) (1992) 1394–1402.
- [14] J. Brown, M. Puckette, A high resolution fundamental frequency determination based on phase changes of the Fourier transform, *J. Acoust. Soc. Am.* 94 (1993) 662.
- [15] G. Costantini, R. Perfetti, M. Todisco, Event based transcription system for polyphonic piano music, *Signal Process.* 89 (9) (2009) 1798–1811.
- [16] G. Costantini, M. Todisco, R. Perfetti, R. Basili, D. Casali, SVM based transcription system with short-term memory oriented to polyphonic piano music, in: *Proc. 15th IEEE Mediterranean Electrotechnical Conference, MELECON, 2010*, pp. 196–201.
- [17] A. Sterian, Model-based segmentation of time–frequency images for musical transcription, Ph.D. dissertation, The University of Michigan, 1999.
- [18] D. Childers, D. Skinner, R. Kemerait, The cepstrum: A guide to processing, *Proc. IEEE* 65 (10) (2005) 1428–1443.
- [19] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, S. Sagayama, Spectrum analysis of polyphonic music signals, *IEEE Trans. Audio, Speech Language Process.* 16 (3) (2008) 639–650.
- [20] H. Olson, *Music, Physics and Engineering*, Dover Publications, 1967.
- [21] M. Ryyänänen, A. Klapuri, Polyphonic music transcription using note event modeling, in: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005*, pp. 319–322.
- [22] S. Abdallah, M. Plumbley, Sparse coding of music signals, Department of Electronic Engineering, King's College London, Tech. Rep., 2001.
- [23] M. Plumbley, S. Abdallah, J. Bello, M. Davies, G. Monti, M. Sandler, Automatic music transcription and audio source separation, *Cybern. Syst.* 33 (6) (2002) 603–627.
- [24] S.A. Abdallah, M.D. Plumbley, Unsupervised analysis of polyphonic music by sparse coding, *IEEE Trans. Neural Netw.* 17 (1) (January 2006) 179–196.
- [25] P. Smaragdakis, J. Brown, Non-negative matrix factorization for polyphonic music transcription, in: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003*, pp. 177–180.
- [26] E. Vincent, X. Rodet, Music transcription with ISA and HMM, in: *Proc. ICA, 2004*, pp. 1197–1204.
- [27] T. Virtanen, Sound source separation using sparse coding with temporal continuity objective, in: *Proc. International Computer Music Conference (ICMC), vol. 3, 2003*, pp. 231–234.
- [28] S. Raczynski, N. Ono, S. Sagayama, Multipitch analysis with harmonic non-negative matrix approximation, in: *Proc. International Conference on Music Information Retrieval (ISMIR), 2007*, pp. 381–386.
- [29] R. Hennequin, B. David, R. Badeau, Score informed audio source separation using a parametric model of non-negative spectrogram, in: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, IEEE, New York, 2011*, pp. 45–48.
- [30] S. Ewert, M. Müller, Using score-informed constraints for NMF-based source separation, in: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, 2012*.
- [31] E. Vincent, N. Bertin, R. Badeau, et al., Two nonnegative matrix factorization methods for polyphonic pitch transcription, 2007.
- [32] E. Vincent, N. Berlin, R. Badeau, Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription, in: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, IEEE, New York, 2008*, pp. 109–112.
- [33] E. Vincent, N. Bertin, R. Badeau, Adaptive harmonic spectral decomposition for multiple pitch estimation, *IEEE Trans. Audio, Speech Language Process.* 18 (3) (2010) 528–537.
- [34] M. Elad, *Sparse and Redundant Representations – From Theory to Applications in Signal and Image Processing*, Springer, 2010.
- [35] J. Brown, Calculation of a constant Q spectral transform, *J. Acoust. Soc. Am.* 89 (1) (1991) 425–434.
- [36] J. Tropp, Greed is good: Algorithmic results for sparse approximation, *IEEE Trans. Inform. Theory* 50 (10) (2004) 2231–2242.
- [37] Y. Pati, R. Rezaifar, P. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: *Proc. 27th Asilomar Conference on Signals, Systems and Computers, November 1993*, pp. 40–44.
- [38] E. Candes, D. Donoho, Curvelets: A surprisingly effective nonadaptive representation for objects with edges, in: A. Cohen, C. Rabut, L. Shumaker (Eds.), *Curve and Surface Fitting*, Vanderbilt University Press, Nashville, TN, 1999.
- [39] E. Candes, L. Demanet, D. Donoho, L. Ying, Fast discrete curvelet transforms, *Multiscale Model. Simul.* 5 (3) (2007) 861–899.
- [40] M. Do, M. Vetterli, The contourlet transform: An efficient directional multiresolution image representation, *IEEE Trans. Image Process.* 14 (12) (December 2005) 2091–2106.
- [41] M. Do, M. Vetterli, Contourlets: A new directional multiresolution image representation, in: *Proc. 26th Asilomar Conference on Signals, Systems and Computers, vol. 1, 2002*, pp. 497–501.
- [42] S. Mallat, Z. Zhifeng, Matching pursuits with time–frequency dictionaries, *IEEE Trans. Signal Process.* 41 (12) (December 1993) 3397–3415.
- [43] K. Engan, S. Aase, J. Husøy, Multi-frame compression: Theory and design, *Signal Process.* 80 (10) (2000) 2121–2140.
- [44] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Trans. Signal Process.* 54 (11) (November 2006) 4311–4322.
- [45] A. Noll, Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate, in: *Proc. Symposium on Computer Processing in Communication, 1969*, pp. 779–798.
- [46] S. Dixon, On the computer recognition of solo piano music, in: *Proc. Australasian Computer Music, July 2000*, pp. 31–37.
- [47] K. Schutte, MATLAB/GNU Octave scripts to read and write MIDI files. [Online]. Available: <http://www.kenschutte.com/midi>.
- [48] R. Rubinstein, KSVD-box. [Online]. Available: <http://www.cs.technion.ac.il/~ronrubin/software.html>.
- [49] B. Krueger, Classical piano midi page. [Online]. Available: <http://www.piano-midi.de>.
- [50] G. Poliner, Data for automatic piano transcription. [Online]. Available: <http://labrosa.ee.columbia.edu/projects/piano>.
- [51] R. Young, Inharmonicity of plain wire piano strings, *J. Acoust. Soc. Am.* 24 (4) (July 1952) 446–458.
- [52] V. Valimaki, M. Laurson, C. Erkut, Commuted waveguide synthesis of the clavichord, *Comput. Music J.* 27 (1) (2003) 71–82.

Michal Genussov received the B.Sc. degree (Cum Laude) in Biomedical engineering in 2008, and the M.Sc. degree in Electrical engineering, in the field of audio signal processing, in 2011, both from the Technion, Israel Institute of Technology. Currently she works as an image processing algorithm engineer in Digital Optics Corporation, a company which develops miniaturized cameras, mainly for the camera phones industry.

Israel Cohen is an Associate Professor of electrical engineering at the Technion – Israel Institute of Technology, Haifa, Israel. He received the B.Sc. (Summa Cum Laude), M.Sc. and Ph.D. degrees in Electrical Engineering from the Technion – Israel Institute of Technology, in 1990, 1993 and 1998, respectively.

From 1990 to 1998, he was a Research Scientist with RAFAEL Research Laboratories, Haifa, Israel Ministry of Defense. From 1998 to 2001, he was a Postdoctoral Research Associate with the Computer Science Department, Yale University, New Haven, CT. In 2001 he joined the Electrical Engineering Department of the Technion. His research interests are statistical signal processing, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification and adaptive filtering. He is a coeditor of the Multichannel Speech Processing section of the *Springer Handbook of Speech Processing* (Springer, 2008), a coauthor of *Noise Reduction in Speech Processing* (Springer, 2009), a coeditor of *Speech Processing in Modern Communication: Challenges and Perspectives* (Springer, 2010), and a general co-chair of the 2010 International Workshop on Acoustic Echo and Noise Control (IWAENC).

Dr. Cohen is a recipient of the Alexander Goldberg Prize for Excellence in Research, and the Muriel and David Jacknow award for Excellence in Teaching. He served as Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS,

and as Guest Editor of a special issue of the *EURASIP Journal on Advances in Signal Processing* on Advances in Multimicrophone Speech Processing and a special issue of the *Elsevier Speech Communication Journal* on Speech Enhancement.